Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 13.11.2023 18:00:24
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____/Д.Г.Демидов/
«18» мая 2022

Рабочая программа дисциплины

«Архитектура и дизайн программного обеспечения»

Направление подготовки
09.03.01 Информатика и вычислительная техника

Профиль
«Системная и программная инженерия»

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2022 г.

Разработчик(и):

старший преподаватель кафедры «Информационная безопасность»

/А.Ю. Гневшев/

Согласовано:
И.о. заведующего кафедрой «Информационная безопасность»,

/А.Ю. Гневшев/

Руководитель образовательной программы,

/А.Ю. Гневшев/

Заведующий кафедрой «Инфокогнитивные технологии»,

доцент, к.т.н. /Е.А.Пухова/

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цель изучения дисциплины – овладеть знаниями и навыками, необходимыми для разработки эффективных и надежных программных систем. В рамках этой дисциплины изучаются принципы проектирования программного обеспечения, архитектурные шаблоны, методы анализа и проектирования систем, а также принципы разработки масштабируемых и поддерживаемых программных решений.

Обучение по дисциплине «Архитектура и дизайн программного обеспечения» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к базовой части основной профессиональной образовательной программы бакалавриата.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Управление разработкой программного обеспечения;
- Анализ и реверс инжиниринг программных средств.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, т.е. 144 академических часа (из них 90 часов – самостоятельная работа студентов и 54 часа – аудиторные занятия).

Разделы дисциплины изучаются в четвертом семестре обучения, т.е. на втором курсе. Форма контроля – экзамен.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			4	
1	**Аудиторные занятия**	**54**	54	
	В том числе:			
1.1	Лекции	18	18	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	36	36	
2	**Самостоятельная работа**	**90**	90	
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/Экзамен		Диф.зачет	
	Итого:	**144**	144	

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1	Принципы проектирования	32	4		8		20
2	Архитектурные шаблоны	34	6		8		20
3	Проектирование систем	34	4		10		20
4	Управление качеством программного обеспечения	44	4		10		30
Итого		**144**	**18**		**36**		**90**

3.3 Содержание дисциплины

1. Введение в архитектуру и дизайн программного обеспечения
2. Принципы проектирования программного обеспечения
3. Модульность и разделение ответственности
4. Повторное использование кода и компонентов
5. Архитектурные шаблоны и стили
6. Клиент-серверная архитектура
7. Многоуровневая архитектура
8. Шаблон MVC (Model-View-Controller)
9. Проектирование интерфейсов пользователя
10. Диаграммы классов и объектов
11. Диаграммы последовательности и взаимодействия
12. Диаграммы компонентов и развертывания
13. Проектирование баз данных в контексте разработки архитектуры сложного программного обеспечения
14. Архитектура сетевых и распределенных систем
15. Архитектурные стили корпоративных приложений
16. Сервис-ориентированные архитектуры
17. Облачные архитектуры
18. Тестирование и отладка программного обеспечения
19. Обеспечение качества программного обеспечения
20. Управление изменениями и конфигурацией
21. Разработка масштабируемых и поддерживаемых систем
22. Работа в команде и коллаборативная разработка
23. Тенденции и новые направления в архитектуре программного обеспечения

4 Учебно-методическое и информационное обеспечение

1. Основная литература

1. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/520097
2. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 147 с. — (Высшее образование). — ISBN 978-5-534-09172-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/513696

2. Дополнительная литература

1. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 248 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/534336

5 Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Практические занятия (семинары) и самостоятельная работа студентов должна проводиться в специализированных аудиториях с комплектом мультимедийного оборудования и/или доской для записей материалов. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

5.2 Требования к программному обеспечению

Для проведения практических занятий (семинаров) специального программного обеспечения для освоения дисциплины не требуется.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, семинары и практики.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;

- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- экзамен.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения				
ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины.	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.

системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.				

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: дифференциальный зачет.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом

по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине.

Шкала оценивания	Описание
Отлично	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1 Экзаменационные вопросы

1. Что такое архитектура программного обеспечения?
2. Какие принципы проектирования программного обеспечения вы знаете?
3. Что такое модульность и как она связана с архитектурой программного обеспечения?
4. Какие архитектурные шаблоны вы знаете и для чего они используются?

5. Что такое клиент-серверная архитектура и как она работает?
6. Какие архитектурные стили используются в корпоративных приложениях?
7. Что такое сервис-ориентированная архитектура (SOA)?
8. Какие преимущества и недостатки облачных архитектур?
9. Что такое анализ требований и как он связан с архитектурой программного обеспечения?
10. Какие диаграммы используются для визуализации архитектуры программного обеспечения?
11. Какие методы тестирования программного обеспечения вы знаете?
12. Что такое обеспечение качества программного обеспечения и как оно связано с архитектурой?
13. Как управлять изменениями и конфигурацией в архитектуре программного обеспечения?
14. Как разрабатывать масштабируемые и поддерживаемые системы?
15. Какие новые тенденции и направления существуют в архитектуре программного обеспечения?
16. Какие основные принципы модульности в программировании?
17. Что такое архитектурный стиль и как он влияет на разработку программного обеспечения?
18. Какие преимущества и недостатки имеет многоуровневая архитектура?
19. Что такое шаблон проектирования MVC и как он применяется в разработке программного обеспечения?
20. Какие методы анализа требований используются при проектировании программного обеспечения?
21. Что такое диаграмма классов и как она помогает в проектировании программного обеспечения?
22. Какие принципы разработки интерфейсов пользователя следует учитывать при проектировании программного обеспечения?
23. Что такое диаграмма последовательности и как она помогает в проектировании программного обеспечения?
24. Какие принципы тестирования программного обеспечения следует учитывать при разработке архитектуры?
25. Что такое обеспечение качества программного обеспечения и какие методы используются для его достижения?
26. Какие инструменты используются для управления изменениями и конфигурацией в архитектуре программного обеспечения?
27. Какие принципы разработки масштабируемых и поддерживаемых систем следует учитывать при проектировании архитектуры?
28. Какие методы коллаборативной разработки программного обеспечения существуют?
29. Какие новые технологии и подходы в архитектуре программного обеспечения стали популярными в последнее время?
30. Что такое микросервисная архитектура и как она отличается от других архитектурных стилей?
31. Какие принципы безопасности следует учитывать при проектировании архитектуры программного обеспечения?
32. Что такое архитектура сетевых и распределенных систем и как она влияет на проектирование программного обеспечения?
33. Какие принципы разработки мобильных приложений следует учитывать при проектировании архитектуры?
34. Что такое архитектура реактивных систем и как она применяется в разработке программного обеспечения?

35. Какие принципы разработки игровых приложений следует учитывать при проектировании архитектуры?
36. Что такое архитектура интернета вещей (IoT) и как она влияет на проектирование программного обеспечения?
37. Какие принципы разработки веб-приложений следует учитывать при проектировании архитектуры?
38. Что такое архитектура блокчейна и как она влияет на проектирование программного обеспечения?
39. Какие принципы разработки искусственного интеллекта следует учитывать при проектировании архитектуры?
40. Что такое архитектура машинного обучения и как она влияет на проектирование программного обеспечения?
41. Какие принципы разработки робототехнических систем следует учитывать при проектировании архитектуры?
42. Что такое архитектура виртуализации и как она влияет на проектирование программного обеспечения?
43. Какие принципы разработки автономных систем следует учитывать при проектировании архитектуры?
44. Что такое архитектура графических приложений и как она влияет на проектирование программного обеспечения?
45. Какие принципы разработки систем управления базами данных следует учитывать при проектировании архитектуры?
46. Что такое архитектура распределенных вычислений и как она влияет на проектирование программного обеспечения?
47. Какие принципы разработки систем машинного зрения следует учитывать при проектировании архитектуры?
48. Что такое архитектура систем управления проектами и как она влияет на проектирование программного обеспечения?
49. Какие принципы разработки систем управления контентом следует учитывать при проектировании архитектуры?
50. Что такое архитектура систем управления процессами и как она влияет на проектирование программного обеспечения?

7.3.2 Типовые практические задания

1. Разработка архитектуры для веб-приложения, учитывая требования к масштабируемости и безопасности.
2. Проектирование архитектуры для распределенной системы, включая выбор протоколов и механизмов коммуникации между компонентами.
3. Создание диаграмм классов и последовательностей для системы управления базами данных.
4. Разработка архитектуры для мобильного приложения, учитывая ограничения ресурсов и требования к производительности.
5. Проектирование архитектуры для системы обработки больших данных, включая выбор подходящих технологий и алгоритмов.
6. Создание диаграмм компонентов и развертывания для системы электронной коммерции.
7. Разработка архитектуры для системы управления проектами, включая модули для планирования, отслеживания и отчетности.
8. Проектирование архитектуры для системы управления контентом, учитывая требования к гибкости и масштабируемости.

9. Создание диаграмм состояний и активностей для системы автоматизации процессов бизнеса.
10. Разработка архитектуры для системы мониторинга и управления сетевой инфраструктурой.

7.3.3 Типовой экзаменационный билет

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1
по дисциплине
«Архитектура и дизайн программного обеспечения»
направление подготовки 09.03.01 Информатика и вычислительная техника

ВОПРОСЫ:

1. Что такое микросервисная архитектура и как она отличается от других архитектурных стилей?
2. Какие методы коллаборативной разработки программного обеспечения существуют?
3. Что такое сервис-ориентированная архитектура (SOA)?

Утверждено: _____ / _____ / «__» _____ 20__г.